# Online Learning-based Robust Visual Tracking for Autonomous Landing of Unmanned Aerial Vehicles

Changhong Fu , Adrian Carrio , Miguel A. Olivares-Mendez , Pascual Campoy

## I. INTRODUCTION

Autonomous landing is a well researched issue for Unmanned Aerial Vehicles (UAVs). In the literature, different kinds of sensing and state estimation technologies/devices have been applied for autolanding, including Differential GPS (DGPS) systems, Laser Range Finders (LRFs), RGB-D sensors, stereo cameras and monocular cameras. However, the DGPS signal can be affected due to the multiple path issues or international jamming, and also prones to lose its accurate position estimation in the urban canyons or flights closing to the ground. The LRFs and RGB-D have the limitation of perception distances, and most of them are still heavy and require more power consumption for Mini-UAVs. And the performance of stereo cameras will be reduced if the baseline is much smaller than the distance between UAV and helipad/scenarios. Hence, the monocular cameras as the most popular and competitive tools are studied for UAV in autolanding tasks recently.

To build a visual tracking algorithm using monocular camera for autonomous landing of UAVs, three main requirements should be included: (I) **Robustness**: it means that the tracking algorithm should be capable of following the helipad accurately even under challenging conditions, such as the significant appearance change, variant surrounding illumination, partial helipad occlusion, rapid pose variation and cluttered background environment. (II) **Adaptivity**: it requires a reliable and sustained online adaptation scheme/mechanism to update/learn the real appearance of helipad. (III) **Real time**: it needs the tracking algorithm to process live video frames with a high speed and performance, thereby generating the consecutive and fast feedback signals for visual controller.

S. Saripalli et al [1] designed and implemented a real-time, vision-based landing algorithm for an autonomous helicopter, which used the moment descriptors to determine the location and orientation of the landing pad, however, it is difficult to apply this visual algorithm in variant outdoor environments, because the intensity values vary significantly depending on sunlight, orientation of the camera, heading of the helicopter and so on. Moreover, it does not have the adaptive characteristic for appearance changes of helipad.

I. F. Mondragon et al [2] also presented a visual tracking algorithm, based on the Lucas-Kanade optical flow, for UAV to land on a helipad, where, the 3D pose of UAV is estimated using a pre-defined reference helipad selected on the first frame, therefore, this tracker also can not learn/update the appearance of tracking helipad, and the RANSAC requires a big number of iterations (heavy time consumption) to reach optimal estimation. Similarily, the SIFT, SURF, ORB, FAST, BRISK feature descriptors are also used in the visual algorithms for autolanding of UAVs, which are referred to the *feature-based* visual tracking approaches.

C. Martinez et al [3] utilized the *direct method* (i.e. directly represent the helipad using the intensity information of all pixels) to track helipad for UAV. They have proved that their tracker performs better than those well-known *feature-based* algorithms and obtained promising results, but it also employed a fixed template (i.e. helipad) during the whole tracking process. Although this tracker has been improved in [4] by manually adding many other templates, but it is not online/self-taught learning. And gradient descent method often falls into a local minimum value and relatively slow close to global minimum.

Our former work [5] applied the adaptively *discriminative method* (i.e. the helipad is separated from its dynamic surrounding background by a adaptive binary classifier, which is online trained/updated with some positive and negative image samples) for UAV to track the helipad, which has obtained the accurate location of helipad. Furthermore, [6] integrated the Multiple-Instance Learning (MIL) approach to improve the robustness of our tracker in more complicated background environments. But both of these trackers can

not provide other state estimations, such as the rotation, scale information of helipad. Even though incorporating these state estimations into the trackers is straightforward, as declared by B. Babenko et al [7] and tested in our tracking experiments, the three performances mentioned above will decrease.

In this paper, motivated by [8], [9], [10], [11], the low-dimensional subspace represention scheme was applied to represent/model the helipad. Additionally, inspired by [4], [7], [12], [13], we adopt the online incremental learning approach to learn/update the appearance of helipad, which has demonstrated good performance to handle the problems of drift, rapid pose variation, variant surrounding illumination and so on. The Particle Filter (PF) [14] rather than gradient descent method was employed to estimate the motion model of helipad. All these changes aim to improve the performances of trackers presented in [3], [4], [5], [6] for autonomous landing of UAVs.

Moreover, we adopt the hierarchical tracking strategy, based on the Multi-Resoution (MR) of frame, to cope with the problems of strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements over time. In addition, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAVs and Ground Control Station (GCS). Using this strategy, especially in the Multi-Particle Filter (MP) voting mechanism, the Multi-Motion Model (MM) will be estimated in the different resolution levels, i.e. the lower resolution features are initially applied to estimate the fewer motion parameters (e.g. location of helipad) at relatively low cost, leaving more motion paramters (e.g. scale and location of helipad) to be estimated in higher resolutions. Besides this mechanism, the Multi-Block Size (MB) adapting method has been utilized to update the helipad with different frequencies, i.e. the smaller (larger) block size means more (less) frequent updates, making it quicker (slower) to model appearance changes and requiring more (less) computation. All these approaches ensured the higher accuracy and real-time performance of helipad tracking.

To the author's best knowledge, this visual tracker has not been presented for solving the online learning and tracking freewill helipad problems in the UAVs, it runs at real-time frame rates and also performs favorably in different autolanding tasks of UAVs in terms of efficiency, accuracy and robustness.

The outline of the paper is organized as follows: In Section II, we introduces the online learning-based visual tracking algorithm based on the low-dimensional subspace represention scheme and online incremental learning approach. Section III proposes the bayes inference model for estimating the motion model, i.e. Particle Filter (PF). Section IV introduces the hierarchical tracking strategy and its configurations. The evaluation of performance results are presented in Section V using aerial image databases from real UAV autolanding flights. Finally, the concluding remarks and future work are proposed in Section VI.

## II. ONLINE LEARNING-BASED VISUAL TRACKING

Online learning-based object tracking has attracted many attentions in recent years, where, those methods via online incremental subspace learning (e.g. G. Li et al [15], T. Wang et al [16], D. Wang et al [17], W. Hu et al [18]) have obtained promising tracking performances. D. Ross et al [19] and F. Yang et al [20] utilized an online incremental learning approach for effectively modelling and updating the tracking target with a low dimensional PCA (i.e. Principal Component Analysis) subspace representation method, which demonstrated that PCA subspace representation with online incremental update is robust to the appearance changes caused by rapid pose variation, variant surrounding illumination and partial target occlusion, as explained by Eq. 1 and shown in Fig. 1. In addition, PCA has also been demonstrated in [11] [21] to have those advantages in tracking applications.

$$\mathbf{O} = \mathbf{U}\mathbf{c} + \mathbf{e} \tag{1}$$

where, $\mathbf{O}$ represents an observation vector, $\mathbf{c}$ indicates the target coding coefficient vector, $\mathbf{U}$ denotes the matrix of column basis vectors, and $\mathbf{e}$ is the error term, which is the Gaussian distribution with small variances.

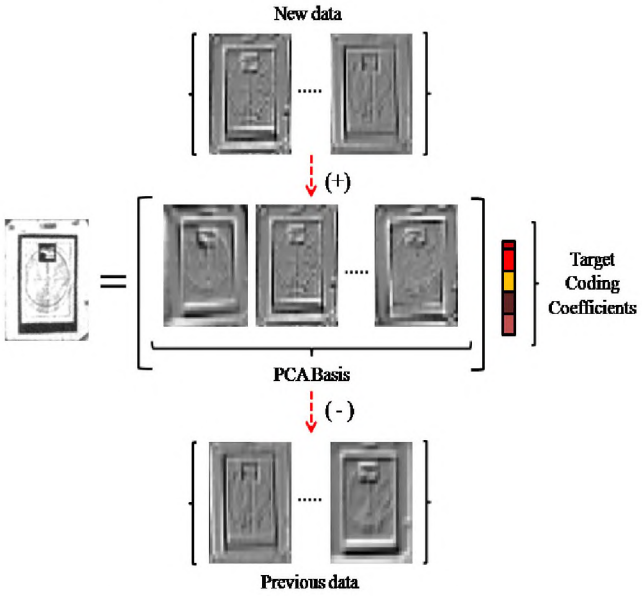



Fig. 1: Incremental PCA subspace learning of a helipad, where, symbol (+) indicates new data are included for updating the appearance of helipad, while symbol (-) means that the previous data are removed from current appearance of helipad.

The main precedures of online incremental PCA subspace learning algorithm with mean update are as follows: Given a set of training image $\mathcal{S}_a = \{\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_n\} \in \mathbb{R}^{d \times n}$, the appearance model of helipad can be computed by the Singular Value Decomposition (SVD) of the centered data matrix $[(\mathbf{S}_1 - \bar{\mathbf{S}}_a) \cdots (\mathbf{S}_n - \bar{\mathbf{S}}_a)]$, denoted by $(\mathcal{S}_1 - \bar{\mathbf{S}}_a)$, i.e. $(\mathcal{S}_a - \bar{\mathbf{S}}_a) = U \Sigma V^\top$, where, $\bar{\mathbf{S}}_a = \frac{1}{n} \sum_{i=1}^{n} \mathbf{S}_i$ is the

sample mean of the training images. If a new set of images $\mathcal{S}_b = \{\mathbf{S}_{n+1}, \mathbf{S}_{n+2}, ..., \mathbf{S}_{n+m}\} \in \mathbb{R}^{d\times m}$ arrives, then the mean vectors of $\mathcal{S}_b$ and $\mathcal{S}_c = [\mathcal{S}_a\, \mathcal{S}_b]$ are computed, i.e. $\bar{\mathbf{S}}_b = \frac{1}{m}\sum_{i=n+1}^{n+m}\mathbf{S}_i$, $\bar{\mathbf{S}}_c = \frac{n}{n+m}\bar{\mathbf{S}}_a + \frac{m}{n+m}\bar{\mathbf{S}}_b$. Because the SVD of $(\mathcal{S}_c - \bar{\mathbf{S}}_c)$ is equal to the SVD of concatenation of $(\mathcal{S}_a - \bar{\mathbf{S}}_a)$, $(\mathcal{S}_b - \bar{\mathbf{S}}_b)$ and $\sqrt{\frac{nm}{n+m}}(\bar{\mathbf{S}}_a - \bar{\mathbf{S}}_b)$, which is denoted as $(\mathcal{S}_c - \bar{\mathbf{S}}_c) = U'\Sigma' V'^{\top}$, this can be done efficiently by the R-SVD algorithm, i.e.:

$$U' = [U\ \tilde{E}]\tilde{U}, \qquad \Sigma' = \tilde{\Sigma} \tag{2}$$

where, $\tilde{U}$ and $\tilde{\Sigma}$ are calculated from the SVD of $R$: $\begin{bmatrix}\Sigma & U^{\top}E \\ 0 & \tilde{E}(E - UU^{\top}E)\end{bmatrix}$, $E$ is the concatenation of $(\mathcal{S}_b - \bar{\mathbf{S}}_b)$ and $\sqrt{\frac{nm}{n+m}}(\bar{\mathbf{S}}_a - \bar{\mathbf{S}}_b)$, $\tilde{E}$ represents the orthogonalization of $E - UU^{\top}E$.

Taking the forgetting factor, i.e. $\eta \in [0, 1]$, into account for balancing between previous and current obserations to reduce the storage and computation requirements, the $R$ and $\bar{\mathbf{S}}_c$ are modified as below:

$$R = \begin{bmatrix}\eta\Sigma & U^{\top}E \\ 0 & \tilde{E}(E - UU^{\top}E)\end{bmatrix} \tag{3}$$

$$\bar{\mathbf{S}}_c = \frac{\eta n}{\eta n + m}\bar{\mathbf{S}}_a + \frac{m}{\eta n + m}\bar{\mathbf{S}}_b \tag{4}$$

where, $\eta = 1$ means that all previous data are included to adapt the appearance changes of helipad.

## III. HELIPAD TRACKING VIA BAYES INFERENCE MODEL

For the autonomous autolanding task of the UAV, the visual helipad tracking can be formulated as an inference problem with a Markov model and hidden state variables. The Particle Filter (PF) [22] is a Bayesian sequential importance sampling technique for estimating the posteriori distribution of state variables characterizing a dynamical system. It provides a convenient framework for estimating and propagating the posteriori probability density function of state variables.

Given a set of observed images $\mathcal{O}_k = \{\mathbf{O}_1, \mathbf{O}_2, ..., \mathbf{O}_k\}$ at the $k$th frame, the hidden state variable $\mathbf{X}_k$ can be estimated as below:

$$p(\mathbf{X}_k|\mathcal{O}_k) \propto p(\mathbf{O}_k|\mathbf{X}_k) \cdot \int p(\mathbf{X}_k|\mathbf{X}_{k-1})p(\mathbf{X}_{k-1}|\mathcal{O}_{k-1})d\mathbf{X}_{k-1} \tag{5}$$

where, $p(\mathbf{X}_k|\mathbf{X}_{k-1})$ is the dynamic (motion) model between two consecutive states, and $p(\mathbf{O}_k|\mathbf{X}_k)$ represents the observation model that estimates the likelihood of observing $\mathbf{O}_k$ at the state $\mathbf{X}_k$. The optimal state of the tracking helipad given all the obserations up to $k$th frame is obtained by the maximum a posteriori estimation over $N$ samples at time $k$ by

$$\widehat{\mathbf{X}}_k = \arg\max_{\mathbf{X}_k^i} p(\mathbf{O}_k^i|\mathbf{X}_k^i)p(\mathbf{X}_k^i|\mathbf{X}_{k-1}), i = 1, 2, ..., N \tag{6}$$

where, $\mathbf{X}_k^i$ is the $i$th sample of the state $\mathbf{X}_k$, and $\mathbf{O}_k^i$ denotes the image patch predicted by $\mathbf{X}_k^i$.

*1) Motion Model:* In this application, we aim to utilize four parameters for constructing motion model $X_k$ of helipad to close the vision control loop: (I) location $x_k$ and $y_k$; (II) scale factor $s_k$; (III) rotation angle $\theta_k$ of the helipad in the image plane, which can be modelled as the $Similarity$ Transformation [23] between two consecutive frames, i.e. $\mathbf{X}_k = (x_k, y_k, s_k, \theta_k)$. The state transition is formulated by random walk:

$$p(\mathbf{X}_k|\mathbf{X}_{k-1}) = \mathcal{N}(\mathbf{X}_k; \mathbf{X}_{k-1}, \mathbf{\Psi}) \tag{7}$$

where, $\mathbf{\Psi}$ is the diagonal convariance matrix, i.e. $\mathbf{\Psi} = (\sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_\theta^2)$. However, a trade off should be found between the efficiency (i.e. how many particles should be generated) and effectiveness (i.e. how well Particle Filter (PF) should approximate the posteriori distribution, which depends on the values in $\mathbf{\Psi}$ ) of PF. Larger values in $\mathbf{\Psi}$ and more particles will obtain the higher accuracy but at the cost of more storage and computation. We solved this problem, i.e. sample improverishment, in the Section IV.

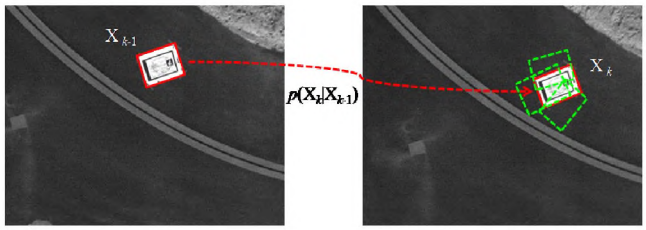



Fig. 2: The dynamical motion model of a tracking helipad, where, the Green Bounding Box represents the test sample generated from Particle Filter, while the Red one is the tracking result with maximum posteriori estimation.

*2) Observation Model:* In this paper, we apply the low-dimensional PCA subspace representation to describe the tracking helipad, thus, a probabilistic interpretation of PCA has been modelled for the image observations. This probability is inversely proportional to the distance from the sample to the reference point (i.e. center) of subspace, which includes two types of distances: $(i)$ the distance-to-subspace: $d_{to}$; $(ii)$ the distance-within-subspace: $d_{within}$.

The probability of $d_{to}$ is defined as:

$$p_{d_{to}}(\mathbf{O}_k|\mathbf{X}_k) = \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, UU^{\top} + \varepsilon I) \tag{8}$$

where, $\boldsymbol{\mu}$ is the center of the subspace, $I$ represents the identity matrix, and $\varepsilon I$ denotes the Gaussian noise.

$$p_{d_{within}}(\mathbf{O}_k|\mathbf{X}_k) = \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, U\Sigma^{-2}U^{\top}) \tag{9}$$

where, $\Sigma$ represents the matrix of singular values corresponding to the columns of $U$.

Hence, the probability of obervation model is constructed as follows:

$$\begin{aligned} p(\mathbf{O}_k|\mathbf{X}_k) &= p_{d_{to}}(\mathbf{O}_k|\mathbf{X}_k)p_{d_{within}}(\mathbf{O}_k|\mathbf{X}_k) \\ &= \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, UU^{\top} + \varepsilon I) \cdot \mathcal{N}(\mathbf{O}_k; \boldsymbol{\mu}, U\Sigma^{-2}U^{\top}) \end{aligned} \tag{10}$$

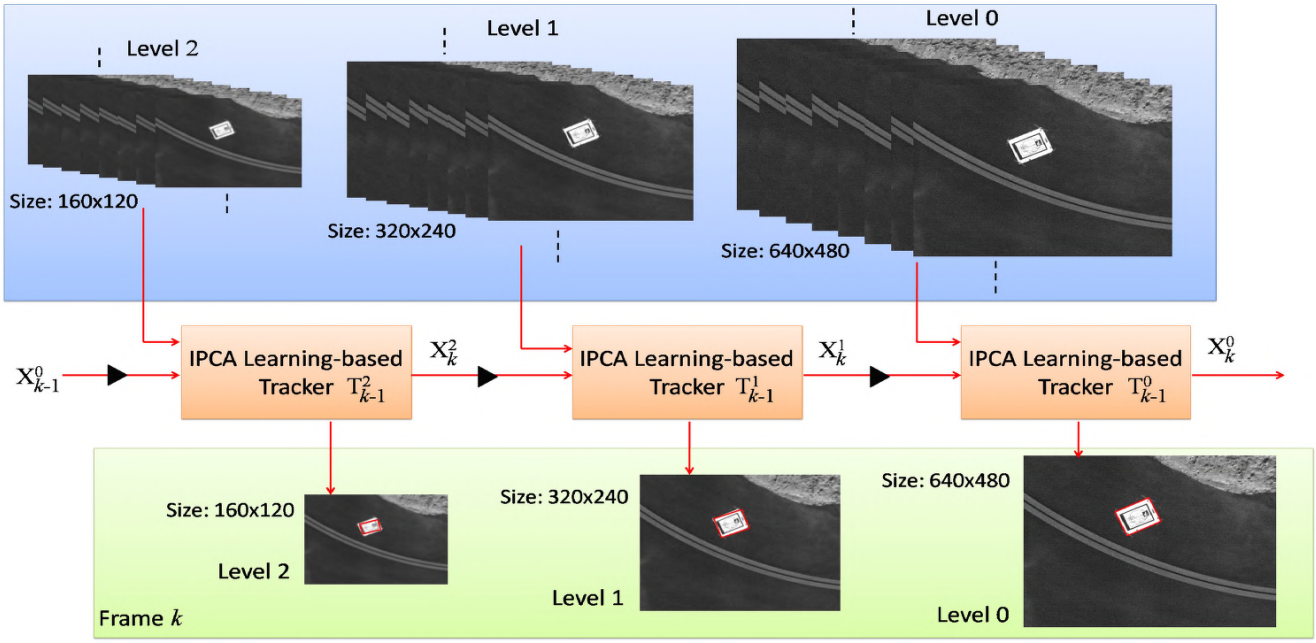



Fig. 3: Our visual tracker. The observation images up to $k$th frame are downsampled to create the MR structure. The motion model estimated in each level is used as the initial estimation of motion for next higher resolution image, leading to a higher tracking speed, better accuracy than a single full resolution-based tracking. The symbol ▶ is the recursion operator.

## IV. HIERARCHICAL TRACKING STRATEGY

In the autolanding application of UAVs, incremental PCA subspace learning-based visual tracker is also sensitive to the strong motions or large displacements. Although the value in $\mathbf{\Psi}$ (in Eq. 7) can be set to be larger, and more particles can be generated to get more tolerance for these problems, however, more noises will be incorporated from those particles, and the requirements of storage and computation cost will be higher, decreasing the real-time and accuracy performances. Therefore, the hierarchical tracking strategy, based on the Multi-Resoution (MR) structure, was proposed to deal with these problems, as shown in Figure 3. The main configurations for hierarchy-based visual helipad tracking are as follows:

### A. Construction of Hierarchical Structure

Considering the image frames are downsampled by a ratio factor 2, the Number of Pyramid Levels ($N_{PL}$) of the MR structure are defined as a function below:

$$N_{PL} = \lfloor log_2 \frac{min\{\mathbf{H}_w, \mathbf{H}_h\}}{minSizes} \rfloor \tag{11}$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $\mathbf{H}_w$, $\mathbf{H}_h$ represent the width and height of helipad $\mathbf{H}$ in the highest resolution image (i.e. the lowest-level of pyramid: 0 level), respectively. And $minSizes$ is the minimum size of helipad in the lowest resolution image (i.e. the highest-level of pyramid: $p_{max}$ level, $p_{max} = N_{PL}$-1), in order to have enough information to estimate the motion model in that level. Thus, if the $minSizes$ is set in advanced, the $N_{PL}$ directly depends on the width/height of tracking helipad $\mathbf{H}$. In this application, the number of pyramid levels is $N_{PL} = 3$, then $p$ is initialized as $p = \{2, 1, 0\}$.

### B. Setup of Multiple Particle Filters

In the Multi-Particle Filter voting mechanism, since the MR structure provides the computational advantage to analyze features and update appearance model in low resolution images, and the lower resolution image is suitable for estimating fewer motion parameters, e.g. location of the helipad, with the increase of resolution, more details from visual information can be used to estimate more parameters of the motion model. In this paper, the motion models estimated in different resolution frames are defined as follows:

Level 2:
$$\mathbf{X}^2_k = (x^2_k, y^2_k), \text{ i.e. } translation$$
Level 1:
$$\mathbf{X}^1_k = (x^1_k, y^1_k, \theta^1_k), \text{ i.e. } translation + rotation$$
Level 0:
$$\mathbf{X}^0_k = (x^0_k, y^0_k, s^0_k, \theta^0_k), \text{ i.e. } similarity$$

where, $k$ is the $k$th frame.

Additionally, the values of $\mathbf{\Psi}$ are smaller in the lower resolution frame, the number of samples generated from related Particle Filter can be also reduced.

### C. Recursion of Multiple Motion Models

Taking into account that the motion model estimated in each level is used as the initial estimation of motion for the next higher resolution image, leaving a higher tracking speed, better accuracy than a single full resolution-based process. The motion model recursion is defined below:

$$x^{p-1}_k = 2x^p_k, y^{p-1}_k = 2y^p_k$$

$$\theta_k^{p-1} = \theta_k^p \qquad (12)$$
$$s_k^{p-1} = s_k^p$$

where, $p$ represents the $p$th level of the pyramid, $p = \{p_{max}, p_{max}-1, ..., 0\} = \{N_{PL}-1, N_{PL}-2, ..., 0\}$, and $k$ is the $k$th frame.

After found the motion model in the $k$th frame, this motion model as the initial estimation is sent to the highest pyramid level of ($k$+1)th frame:

$$x_{k+1}^{p_{max}} = \frac{x_k^0}{2^{p_{max}}}, y_{k+1}^{p_{max}} = \frac{y_k^0}{2^{p_{max}}}$$
$$\theta_{k+1}^{p_{max}} = \theta_k^0 \qquad (13)$$
$$s_{k+1}^{p_{max}} = s_k^0$$

where, the $2^p$ and $\frac{1}{2^p}$ are called recursion operator ($\blacktriangleright$), as shown in the Figure 3.

### D. Propagation of Multiple Block Sizes

As introduced in the Section I, the image in the highest (lowest) of pyramid has less (more) texture information, thus, the Multi-Block Size (MB) adapting method has been utilized to update the helipad with different frequencies/speeds, i.e. the smaller (larger) block size means more (less) frequent updates, making it quicker (slower) to model appearance changes and requiring more (less) computation. The propagation of block size ($N_B$) is given as below:

$$N_B^{p-1} = \lfloor \frac{N_B^p}{log_2(2+p)} \rfloor \qquad (14)$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $p$ represents the $p$th level in the pyramid, and $k$ is the $k$th frame.

## V. EVALUATION RESULTS

In this section, we compare our visual tracker with Ground Truth databases in two different UAV autolanding flight tasks. The original frame size is 640×480, the ROS[1] framework has been used to manage and process image data.

### A. Ground Truth Collections

Ground Truth (GT) databases are applied to analyze the performance of our visual tracker. Figure 4 shows the ground truth points, which will be zoomed in and clicked by mouse to obtain the location of each point. The center location, rotation and area of helipad can be calculated frame-to-frame based on these point locations.

### B. Results and Comparisions with Ground Truth

*1) Test 1:* In this test, it contains three main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Rapid pose variation; (III) Illumination Variation.

Some tracking results in Test 1 are shown in Figure 5.

The comparison with Ground Truth is shown in Figure 6 and 7, where, the average RMSE of X, Y position, Rotation and Area are 2 Pixels, 3 Pixels, 2 Degrees and 133 Pixel$^2$ (i.e. ∼2cm in Height), respectively.

[1]http://www.ros.org/

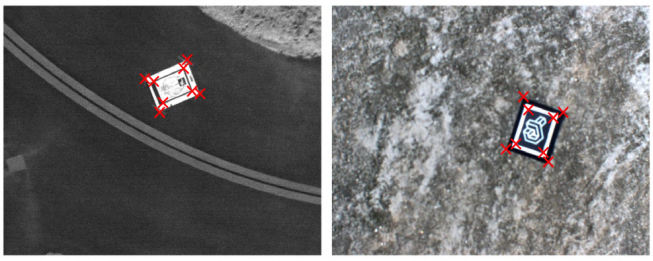
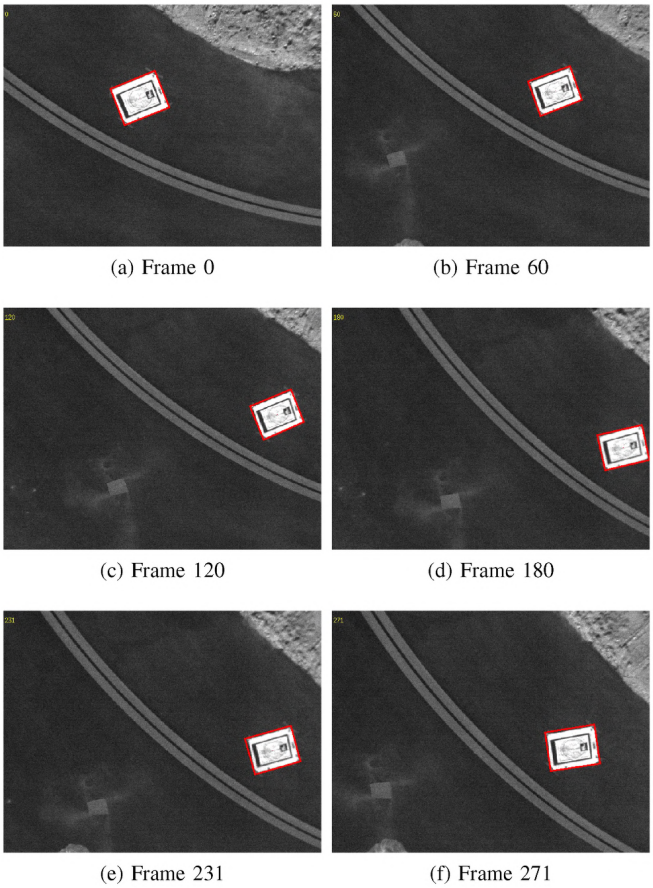


Fig. 4: Ground Truth Points. The corners, as the obvious features, are selected to establish the GT databases.



(a) Frame 0 (b) Frame 60

(c) Frame 120 (d) Frame 180

(e) Frame 231 (f) Frame 271

Fig. 5: Tracking results using our visual tracker in Test 1.

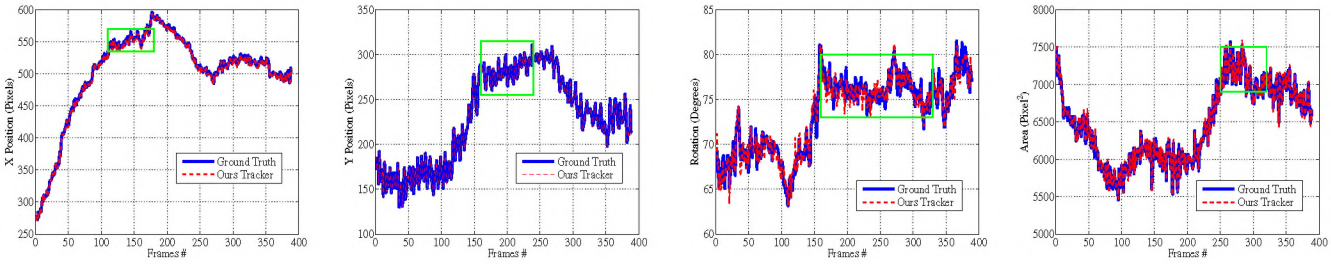
*2) Test 2:* In this test, it consists of four main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Scale change; (III) Illumination Variation; (IV) Rapid pose variation.

The tracking results in the Test 2 are shown in Figure 8.

The comparisons with ground truth are shown in Figure 9 and 10, where, the average RMSE of X, Y position, Rotation and Area are 3 Pixels, 4 Pixels, 3 Degrees and 235 Pixel$^2$ (i.e. ∼3cm in Height), respectively.

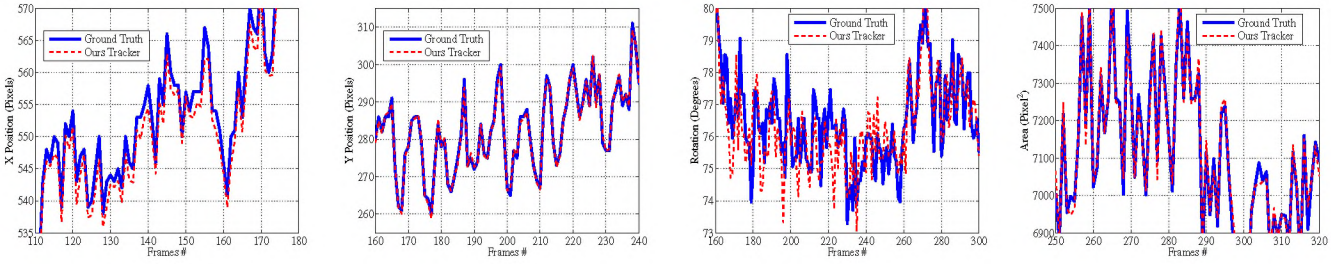
## VI. CONCLUSIONS AND FUTURE WORKS

Considering the limitations of former works, this paper presented a novel robust visual algorithm for UAVs to



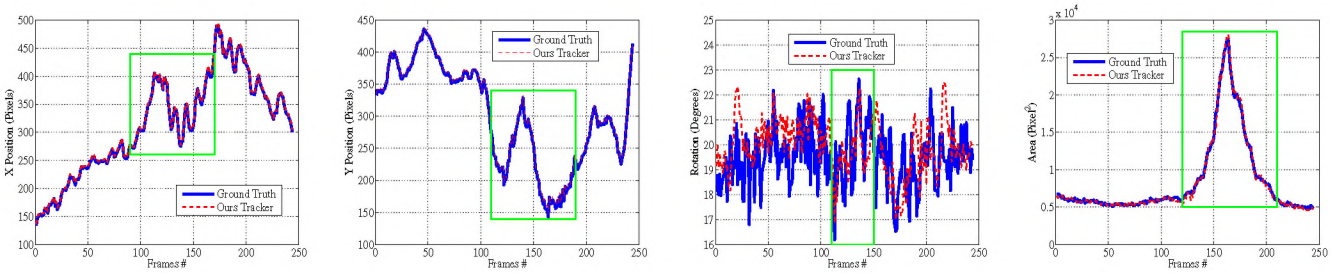

(a) X Position (b) Y Position (c) Rotation (d) Area

Fig. 6: Comparison with Ground Truth in Test 1.



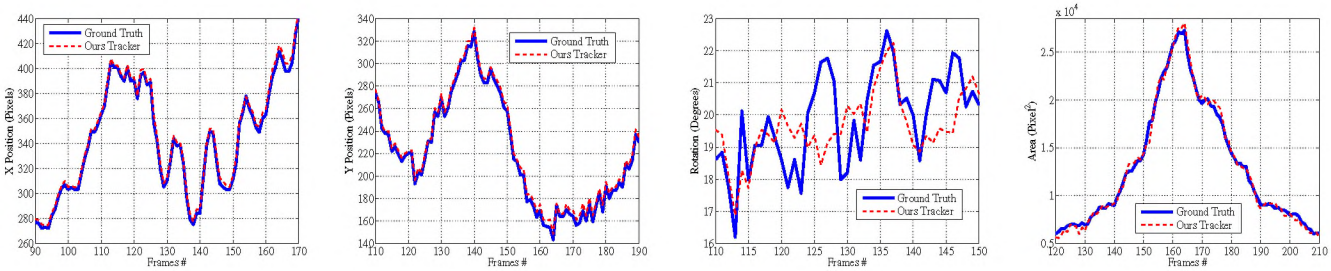

(a) X Position (b) Y Position (c) Rotation (d) Area

Fig. 7: Enlarged Region from Green Rectangle in Fig. 6.




(a) X Position (b) Y Position (c) Rotation (d) Area

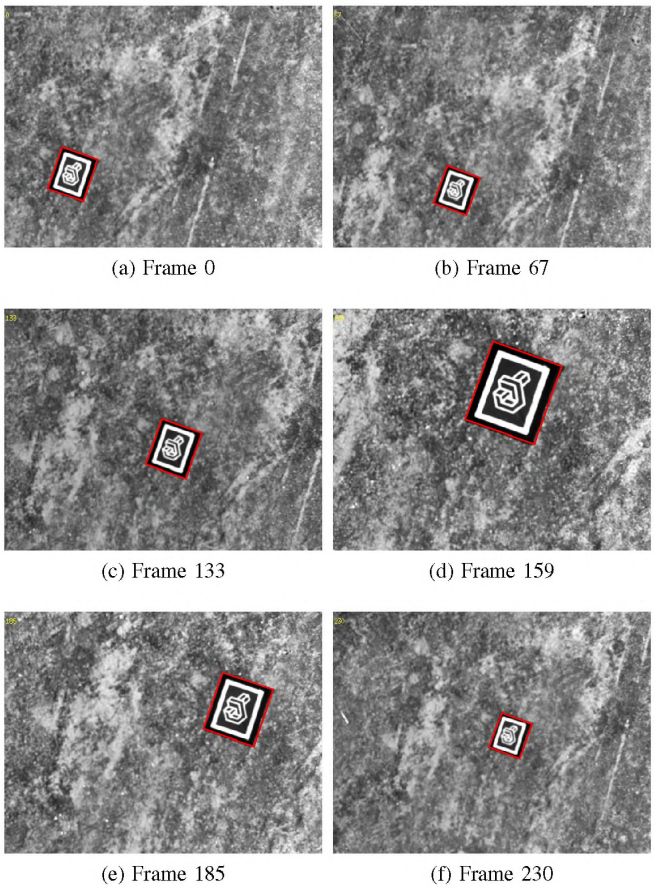
Fig. 9: Comparison with Ground Truth in Test 2.




(a) X Position (b) Y Position (c) Rotation (d) Area

Fig. 10: Enlarged Region from Green Rectangle in Fig. 9.

autonomously land on an freewill cite regardless of challenging situations such as significant appearance change, variant surrounding illumination, partial helipad occlusion, rapid pose variation et al. It integrates the low-dimensional subspace representation scheme, online incremental learning approach and hierarchical tracking strategy to effectively and efficiently estimate the location, rotation and area information of helipad at real-time frame rates of more than twenty



(a) Frame 0 (b) Frame 67

(c) Frame 133 (d) Frame 159

(e) Frame 185 (f) Frame 230

Fig. 8: Tracking results using our visual tracker in Test 2.

frames per second. And compared to the Ground Truth data, it is accurate to close the vision-based control loop for UAVs to carry out more autolanding tests.

In the future works, we will apply the incremental learning method for negative sample (i.e. background information) rather than only positive sample (i.e. helipad) to estimate the state of helipad during the autolanding tasks. And we will compare the IMU/GPS data recorded from the UAVs.

## ACKNOWLEDGMENT

The work reported in this paper is the consecution of several research stages at the Computer Vision Group-Universidad Politécnica de Madrid. This work has been sponsored by the Spanish Science and Technology Ministry under the grant CICYT DPI2010-20751-C02-01, the IRSES project within the Marie Curie Program FP7 (UECIMUAVS - USA and Europe Cooperation in Mini UAVs) and the China Scholarship Council (CSC). And authors would like to thank the reviewers for their valuable feedback and input.